

Service-based Enterprise Application Integration with Message-oriented Middleware

Chen Rong

Information Center China Three Gorges Corporation, Yichang City, China
chenr@ctgpc.com.cn

Abstract—Information sharing and collaboration brings about interest of Enterprise Application Integration (EAI) for “islands of information coherence” in the organization. EAI actually can be an elastic framework based on Service-oriented Architecture (SOA), developed from the bottom up by means of existing technologies. This paper introduces Message-oriented Middleware (MOM) with detail explanation on its implementation framework and operational principles, and proposes an approach to build a flexible and reliable messaging system based on MOM. Both SOAP and database adapters provided by MOM are utilized to store-and-forward all the todo tasks diversified in different applications. Users can access the relevant todo information in Enterprise Information Portal with great improvement on the operation efficiency, instead of visiting different application systems to deal with the todo tasks.

Keywords—service-oriented architecture; enterprise application integration; message-oriented middleware; SOAP

I. INTRODUCTION

The world is increasingly becoming more connected in many different ways, and the amount of information we have at our fingertips seems to have exploded. As the speed and efficiency of the underlying infrastructure improves, so does the organization's ability to control and take advantage of information assets through information sharing and collaboration. However, the more we share information, the more we realize that years of distribution of computing power and business applications across the different lines of business has led to “islands of information coherence.” Therefore, the combination of information sharing and operational collaboration becomes a differentiating factor for effective management of the business and, ultimately, to competitive advantage.

Meanwhile, the greater challenge here is that the various systems that need to be linked together often reside on different operating systems, use different database solutions and different computer languages, and in some cases are legacy systems that are no longer supported by the vendor who originally created them. In some cases, such systems are dubbed “stovepipe systems” because they consist of components that have been jammed together in a way that makes it very hard to modify them in any way [1].

This brings about interest in Enterprise Application Integration (EAI), which is the process of linking different applications together within a single organization or across organization boundaries in order to simplify and automate business processes to the greatest extent possible, while at the same time avoiding having to make changes to the existing applications or data structures. In the words of the Gartner Group, EAI is the unrestricted sharing of data and business

processes among any connected application or data sources in the enterprise [2]. EAI usually involves the data exchange which achieves a uniform data view of participating systems, and the business interaction which accomplishes mutual invocation across boundaries of autonomous systems in real time.

From the perspective of technology, EAI is an integration framework composed of a collection of technologies and services which form a middleware to enable integration of systems and applications across the enterprise. Actually, middleware can be viewed as a reusable, expandable set of services and functions that are commonly needed by many applications to operate well in a networked environment. A Service-oriented Architecture (SOA) is essentially a collection of services, and a flexible set of design principles used during the phases of systems development and integration. A deployed SOA-based architecture will provide a loosely-integrated suite of services that can be used within multiple business domains and reformed with the changing of business operation. The remote data access can be encapsulated in the form of web services and be called from applications. Implement these either synchronously or asynchronously, depending on the requirements of the application. This approach greatly increases the reusability of data to applications, and generally performs and scales quite well. Actually, there are many real cases of service-based EAI have been developed. O. R. Bagheri et al. present the elastic EAI framework which has a service-based architecture and could be developed from the bottom up by means of existing technology [3]. Frequent mentioned service-based EAI cases usually occur in the fields such as Enterprise Information Portal [4].

More and more new technologies are rolling out to improve the flexibility and reliability of the interfaces between systems. Stefan Tai et al. suggest that the use of existing Message-oriented Middleware (MOM) for reliable Web services messaging seems reasonable [5]. Different with the information exchanging technologies in the distributed computing environment, such as RPC (Remote Procedure Call) [6], OTM (Object Transaction Manager) [7], Java Message Service (JMS) compatible Message-oriented Middleware (MOM) does not need reliable data transportation layer. For message queuing systems, they typically provide enhanced resilience functionality to ensure that messages do not get “lost” in the event of a system failure. Normally both synchronous and asynchronous modes are supported by MOM. In asynchronous mode, the source application does not need to send message to the destination at once, MOM will send the message to the appropriate endpoint and only send once time [8]. MOM also reduces the complexity of developing applications that span multiple operating systems and network protocols by insulating

the application developer from the details of the various operating system and network interfaces. With SOA and MOM, the database link in homogeneous environment or other technology like programming on ODBC, JDBC can be instead of mesh service interfaces based on Enterprise Service Bus (ESB).

In this case, two kinds of adaptor - SOAP and database one - are implemented to connect MOM system to different application systems; todo tasks information existing in the workflow processes based structured Oracle Database and unstructured Domino database are collected, stored, presented in the Enterprise Information Portal, the SMS for notice sent to the individual's mobile phone. Based on the universal identification of person and single sign-on (SSO), the integration set of todo tasks, which are presented in the Enterprise Information Portal and assigned to the right individuals, really improves the operation efficiency for users. This could be a very good example of enterprise application integration.

II. IMPLEMENTATION FRAMEWORK OF MOM

MOM simplifies data transportation between applications, insulates the ground heterogeneous operating system and network, and provides uniform communication protocol and programming interface. MOM supports not only the synchronous mode, but also the asynchronous mode. Based on the store-and-forward data transportation, the asynchronous mode could provide better fault-tolerance than the synchronous mode. Data exchanging based on the message transportation and asynchronous transaction processing makes it possible for the application integration in times of SOA.

There are mainly two kinds of asynchronous middleware: broadcasting and publishing/subscribing. Since the publishing/subscribing mode could specify the type of messages for the subscribed users, it has already come to be the de facto standard with well targeted. Examples of commercial implementations of this kind of MOM include IBM's WebSphere MQ (formerly MQ Series), Oracle Advanced Queuing (AQ), MessageQ (formerly BEA's). There is a Java standard called JMS - Java Message Service (formerly brought up by SUN), which has associated with it, a number of implementations, both proprietary and free software, for example, the ActiveMQ from Apache. The implementation of this paper was based on ezESB-MQ, JMS compatible middleware developed by Tongfang Software in mainland China.

A. Implementation Framework of ezESB-MQ

ezESB-MQ follows the JSR 208 and JMS 1.1, and builds an efficient MOM based on message queuing. Its key function is to provide the reliable and securable message transportation for different applications span multiple operating systems and network protocols. As shown Figure 1, the efficient message framework and message service components in ezESB-MQ could provide many good features like clustering, peer-to-peer network, plug-in protocol support (such as HTTP, TCP/SSL). ezESB-MQ also supports the fast persistence through JDBC, and high performance logs. ezESB-MQ can be seamlessly

integrated with the java virtual machine and the java container following the specification -J2EE 1.4.

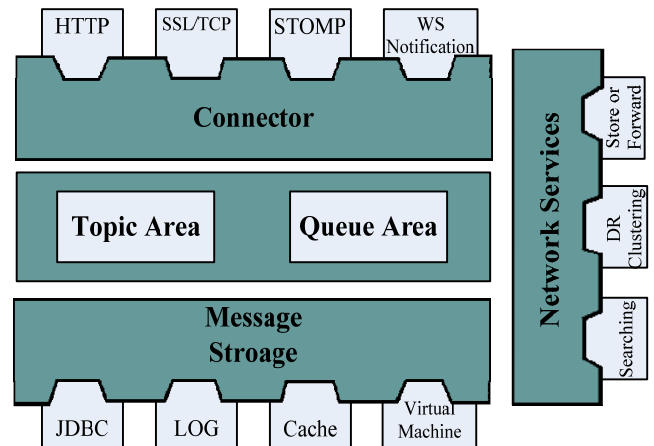


Figure 1. Implementation Framework of ezESB-MQ.

With plugged-in different types of ezDX adaptor, such as database adaptor, SOAP adaptor, and ERP adaptor, the MOM could easily connect to different systems. This makes it possible for users to choose different connection strategies according to different applications. The data from external systems is wrapped into JMS encapsulation format by ezDX adaptor, and at the meantime, MQ is responsible for message routing.

Beyond the MQ, ezESB provides the service bus and management platform for distribution application. For example, the SOAP adaptor could be implemented into a web service and published on the ezESB to be called by external systems. On the other hand, the messages routing also could be exposed as web services on the ezESB to be invoked. By insulating the data access and message routing, the data transferring is totally transparent for the applications, and the users. It is obviously that the ezESB, which is a combination of MOM and service bus, could be the good foundation of enterprise application integration.

B. Operational Principle of ezESB-MQ

The basic operation principle of ezESB-MQ was briefly described in Figure 2.

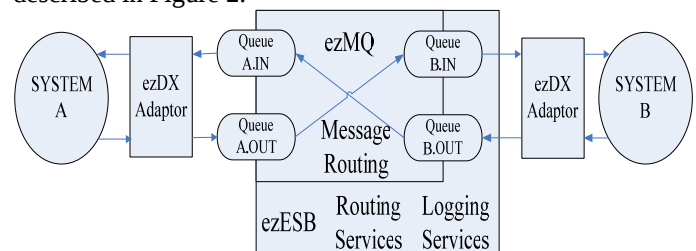


Figure 2. Operational Principle of ezESB-MQ.

In order to accomplish the data exchanging between the system A and B, we can use the MQ and ezDX adaptors to build the connection between the system A and B:

- Firstly, ezDX adaptor could support multiple protocols and use both active and passive method to get data from external system A.
- Secondly, there are only two physical queues existing on ezMQ, one queue for writing message, the other for reading. External system uses peer-to-peer method to transport data with MQ middleware.
- Thirdly, after receiving data from system A, the ezDX adaptor can wrap data into exDX message formatted in XML (specified by XSD document); then, the adaptor wrapped ezDX message into MQ message formatted in JMS and sent it into the output message queue according to system A on the MOM.
- Fourthly, ezDX will periodically poll data from the queue A.IN and send into system A.
- Fifthly, ezMQ will load routing table with system identifier and all the queues IN and OUT information. ezMQ starts to work after ezDX sends messages to the queue A.OUT. ezMQ receives the message from A.OUT, and analyzes the message head information, then deliver the message to system B according to the routing node definition.

Thus it can be seen, the systems which provide the information for sharing do not know the location with each other, and how to build the connection. The systems even do not need to be running at the same time, or running on the same operating system or network protocol. ezESB-MQ can provide loose couple connections between different systems for application integration.

III. CASE STUDY: IMPLEMENTATION OF TODO TASKS INTEGRATION

With the development of enterprise business, more and more applications such as OA, project and engineering management system, financial management system, human resources system, document archiving management system, portal, etc., are put into use. In the process to be planned and implemented during different phases of informatization, these systems are independent from each other, but actually have tightly relationship on data level. In order to break down the island of information, it is necessary to start the enterprise application integration to organize the public information, and make it possible for different business units to work together. As a trial, the relatively popular architecture – SOA and MOM are introduced to integrate diverse systems in the enterprise.

In our heterogeneous and distributed enterprise application environment, the OA system is based on Domino system with third-party workflow engine, a workflow management system is applied in different projects and organizations within the enterprise with different versions and different installations (as EIIS, TGPMS Workflow, XLDPMS Workflow, XJBPMs workflow shown in Figure 3), a document archiving system is developed by SSH (Spring+ Struts + Hibernate). EIIS is the first installation of the workflow management system developed for China Yangtze Power Corporation, which is a public corporation controlled by China Three Gorges

Corporation. The other three PMS Workflow are the same version but much newer than EIIS. EIIS is much more like OA application based on RDBMS, and there are different workflows, such as document, expert, supplier, purchase management workflows running on it. On the other hand, PMS workflow systems manage the document transferring and contract payment processes. In the document archiving system, only one human workflow process there for document leveraging request and approval. Finally, there are three kinds, but totally six systems to be integrated to extract todo tasks and presentation in portal. In the meanwhile, the messages from the particular OA workflow processes need to be sent to SMS Gateway and mobile phone of related person.

A. Topology of Todo Tasks Integration

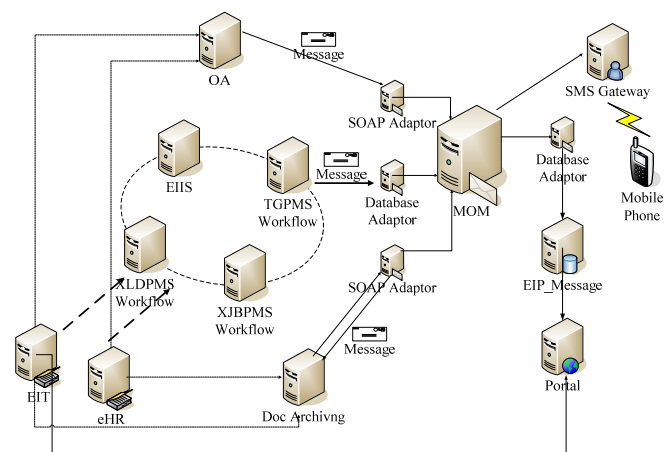


Figure 3. Topology of Todo Tasks Integration.

By research and analysis, we can choose different strategies to make the connection with ezESB-MQ for different application systems depended on implementation circumstance. For example, ezESB exposes the SOAP adaptor as a web service to document archiving system, at the same time, ezESB gets data from workflow management system by the database adaptor using periodically polling method. Actually, the document archiving system is under development stage, and it could be asked to generate the XML format message for todo tasks according to our specification and send to the web service – SOAP Adaptor of MOM, then store into the destination after message converting and routing. On the other hand, the workflow management system is able to organize an accurate view to list todo tasks in detail, then the database adaptor polls the data and compares with the sync table to extract information and sends to destination. For OA system, we try both SOAP and database adaptor for connection. It is rather more complicated in OA system because of the third-party workflow engine and plenty of workflow processes. We need analyze the relationship between every process and the engine to discover the appropriate node which triggered the message delivery. We also need to maintain the common message queue in OA system for different workflow processes, i.e., implement the store-and-forward method in the OA system itself, and asynchronously access the SOAP adaptor resided on ezESB-MQ. Obviously, the transfer mechanism will be more and more

complicated with more and more workflow processes. In order to accomplish the integration, it is not enough to have the message queuing system; at the same time, every application system and portal must reference to the uniform personal indentify information table (EIT) and human resource information (eHR) as shown in Figure 3.

B. Storage Specification of Todo Tasks

After requirement analysis, the integrated todo tasks needs to present all the todo tasks from diverse systems but also separate them into categories of sponsor, operator, and system identifier. Therefore, no matter what kind of middleware is adopted, the enterprise application integration could not be separated from the data integration. Since data integration is the problem of combining data residing at different sources, and provides the user with a unified view of these data [9,10,11], it is very important to give out the well-designed global schema of todo tasks. Also, the mapping between global and local schema is necessary. Based on analysis of the data of todo tasks and workflow processes in every system, the global schema for todo tasks is abstracted and given out as a base table – EIP_MESSAGE, the structure in detail was shown in table I.

C. Importance of Global ID

In Table I, there are several IDs in the table EIP_MESSAGE, but only the userid for sender and receiver as well as the application id are very important. Only few systems or applications there in the organization, it is very easy to maintain the consistent use of the application ID. On the other hand, there are usually different identifiers for the same person in different systems since different systems developed in different stages without overall plan and design. It is the same in this case, therefore in Figure 3, EIT is for ID generation, but EIT depends on the mapping with human resource system (eHR) on the single, unique person to get right personal information. Messages about to do tasks assigned to the person should be transferred to the destination with the global ID linked to the person.

Generally speaking, it is a principle for us to generate global IDs by only one authorized system, so that each value is unique; and, assign global IDs to link and reconcile matched records about the same person from different data sources. Otherwise, it is an impossible task to collect right todo information for the individuals and display them in the portal. EIT system plays this role in the case, which is referenced by all the participating systems.

D. Implementation of SOAP Adaptor

On the side of ezESB-MQ, customized SOAP adaptor could be wrapped as a web service and publish the data interface by WSDL (Web Services Description Language). On the other side, source system always implements the workflow processes by its own workflow engine or record the transaction by state machine. Therefore, as long as the source system has the ability to consume the web service then it has the ability to connect to ezESB-MQ middleware through the SOAP adaptor. Normally, it is better for the developer to design the message delivering classes for calling when the workflow processes or

the transaction status changed. The processing flow for sending message via the SOAP adaptor is shown in Figure 4.

TABLE I. EIP_MESSAGE SPECIFICATION

Field Name	Description	Data Type
Transaction ID	Application ID + Transaction ID, i.e., 0030001	Varchar(256)
Sender	The userid of sender, corresponding to producer in JMS, reference to the uniform person indentify information	Varchar(512)
Receiver	The userid of receiver, corresponding to consumer in JMS, reference to the uniform person indentify information	Varchar(512)
Title	The title of message	Varchar(3000)
Timestamp	Time by sending, i.e., YYYYMMDDhhmiss	Varchar(50)
Application ID	Current application code no, i.e., 003	Varchar(3)
Workflow ID	Current workflow ID, i.e., 8341	Varchar(50)
Workflow Name	Current workflow Name	Varchar(512)
Node ID	Current workflow node ID	Varchar(512)
Node Name	Current workflow node name or description	Varchar(512)
Node Status	status for current workflow node, (RUN/OVER)	Varchar(50)

E. Implementation of Database Adaptor

Database adaptor from ezESB-MQ provides both the file-based and database-based data matching and extraction mechanism. Both of them need the database adaptor

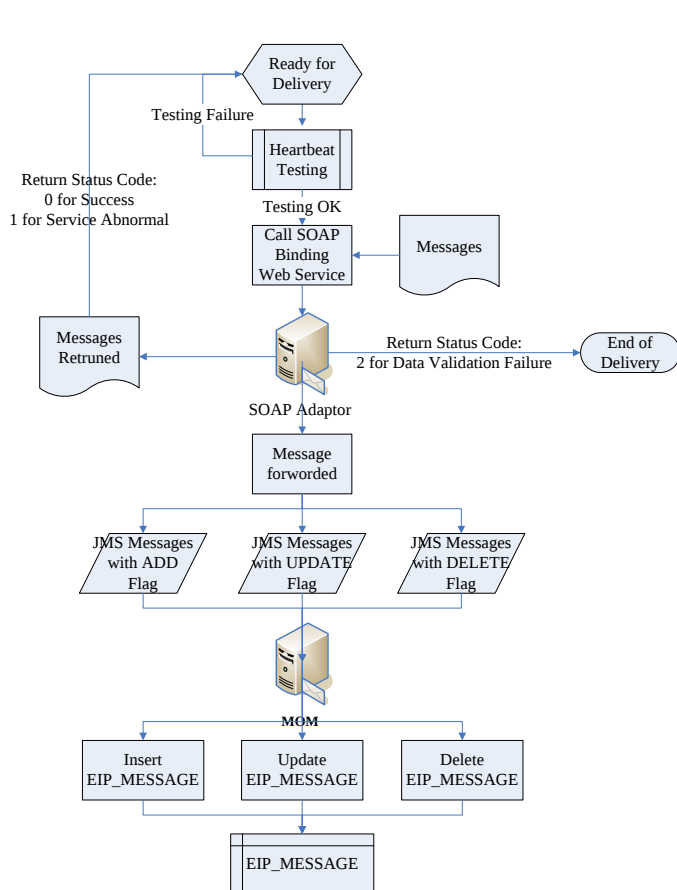


Figure 4. Implementation of SOAP Adaptor.

periodical poll the data from source to get the original data. For file-based matching, the adaptor will store the polling result in file cache temporarily. As well, the database-based matching needs a sync table to store the data. As show in Figure 5, the database adaptor retrieves the data by matching the index value and hash value to decide how to manipulate with the data in EIP-MESSAGE.

Comparing with the index value and the hash value of the whole record content in both source and sync tables, it is easy for database adaptor to determine the specific record from source matching with a record in sync table or not. In case there are records existing in source without index matching records in sync table, additional records should be inserted into sync table; if there are records with same index value but different hash value in source and sync table, change should be made to the records in sync table; otherwise, records in sync table should be deleted. The JMS message will be generated based on the data in sync table, then forwarded to MOM, and finally corresponding records in EIP_MESSAGE are maintained.

F. Unique View of Todo Tasks in Portal

The key function of the todo tasks presentation program,

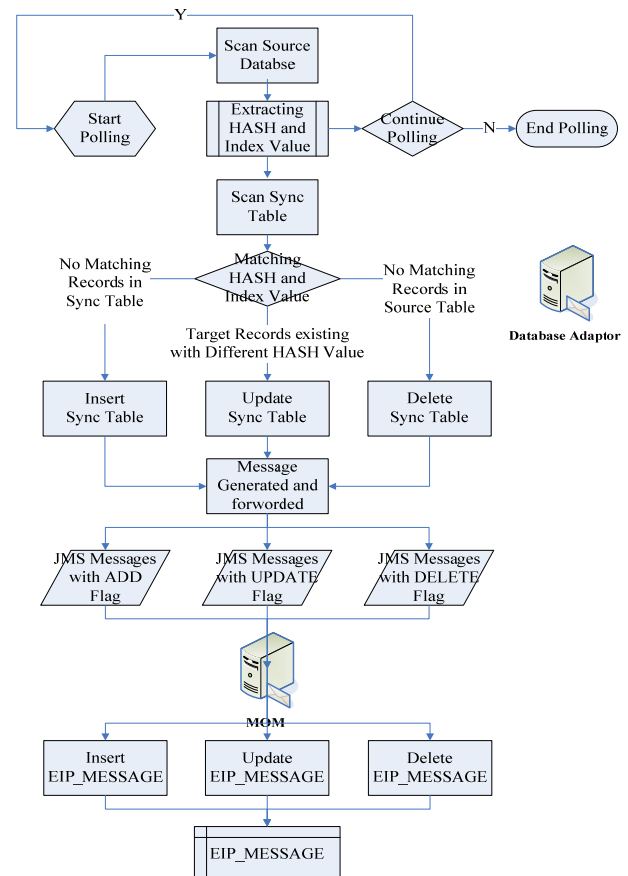


Figure 5. Implementation of Database Adaptor.

which is working as an external component of the enterprise information portal based on Oracle iAS, is to get the Global user ID from portal, and generate the query sql script to retrieve the todo information specified for the exact Global user ID.

It is worthy mention here, the value of the field “node status” in the table EIP_MESSAGE changes over time according to the XML messages received by SOAP adaptor, or the content of sync table. Despite different specification of workflow, it is enough for us to decide which records should be displayed in the column of “TODO” or “DONE” for each user by utilizing two kinds of node status --“RUN”/“OVER”. In our case, “RUN” means the tasks are waiting for processing, while “OVER” means the tasks has been handled.

Hyperlink text is attached to every todo tasks displayed in the portal with the specific URL. It is easy for user to click on the tasks in the “TODO” column, and jump into the relevant workflow system by taking advantage of the single sign-on feature of the portal. As we know, SSO is enabled by Oracle Internet Directory (OID) within the infrastructure of iAS, which stores the credential information for different systems according to the user. With SSO, the login procedure is simplified and user can go through into the relevant workflow process to handle the tasks directly. The processing results in the workflow system will be captured by the adaptor and then be forwarded by MOM to the EIP_MESSAGE. All the processes

are automated, and the operation efficiency is improved by the collecting of todo tasks. On the other hand, with database adaptor, the SMS gateway also store-and-forward the todo information from MOM to users' mobile phone for notice.

The todo tasks information is successfully integrated in the Enterprise Information Portal by the implementation of MOM in 2010, and users benefit from the uniform view of the todo tasks from the participating systems. By taking advantage of Service-based EAI, further automation of business process could be realized by exposing every workflow nodes as the services on the ESB. Users can make the decision, or fill in any information just in the portal instead of handle the task in the separated workflow systems.

IV. CONCLUSION AND ONGOING WORKS

The EAI domain provides integration models and techniques for assembling heterogeneous software applications in a pragmatic way. EAI emerging solutions encompass (1) a distributed architecture using web services and (2) a description of the web services centric architecture [12]. But SOA is not enabled for all-purposes. There are a couple of drawbacks of SOA Web Services [13]:

- Web services are harder and more costly to write, test, and deploy.
- The organization runs the risk of creating an "SOA Nightmare" of numerous point-to-point, application-specific, non-reusable web services, all of which need to be maintained in response to changing database schemas and locations.
- It is difficult for database objects to consume web services. They must usually be consumed by applications. Some of the newer DBMSs permit you to encapsulate application classes as stored procedures or functions; however, this method will not work for views.

Even these could not stop us for choosing SOA and MOM, a formal architecture-centric approach can be taken to formalize, deploy and involve Service-oriented Architecture [12]. But we suffer a lot from the inconsistency data about person across different systems. For effective service delivery on any infrastructure and SOA-based environment, data is an essential building block both as inputs as well as in monitoring the output. Data is an integral part of any formation delivery and/or service delivery in an IT-enabled business environment [14]. Without the correct mapping of person ID in different systems, nor without the "golden record" of person in the enterprise, the todo tasks retrieved from diverse systems could not be put into the todo list in portal for the right person.

Actually, there are some rules in the integration. For example, "There is no common standard", i.e., it is not better for us to have too many software standards than we have no standard [13]. In fact, even the successful internet standard like TCP/IP is not commonly used worldwide. Therefore, the integration team needs to define the internal specification inside the organization, and tries to keep in pace with external industry standards. Actually, we do care about the internal standard. Some rules concluded in the todo tasks integration

based on MOM may guide us to setup internal specification for application integration.

Firstly, follow the top-down design principle to do the requirement analysis; what kind of information need to be collected and put into where should be decided; the business process driven methodology for enterprise integration provides a complete understanding about the business process importance as the central model in an enterprise integration project [15].

Secondly, follow the bottom-up working principle to do the analysis on data level and meet the requirement from business processes interoperability.

Thirdly, the specification for common data (or master data) is the key for integration, collaboration and business intelligence; sometimes the canonical data model should be setup for data exchanging; data governance and data quality control is the foundation of the application integration.

Finally, there is no best technology in the world, only suitable technology in the enterprise application environment. It is better for us to do further research in MOM and SOA utilization.

Generally speaking, it is rather hot to do research for the application integration with modern information technology. It is practical to realize the todo task integration in the framework of the service-based Enterprise Application Integration with the implementation of Message-oriented Middleware. The operation efficiency is greatly improved by providing central access of all the todo tasks in the Enterprise Information Portal. It is very convenient for users to know his/her relevant tasks instead of visiting different application systems. But in the other hand, consolidation is better than integration with lower cost and more efficient [16].

REFERENCES

- [1] David Loshin, "Master Data Management", MK/OMG Press, pp 1,2009.
- [2] Julie Gable, "Enterprise application integration", Information Management Journal, March/April, 2002.
- [3] O. R. Bagheri, R. Nasiri, M. H. Peyravi, and P.Khosraviyan Dehkordi, "Toward an elastic service based framework for Enterprise Application Integration", Fifth International Conference on Software Engineering Research, Management and Applications, 2007, pp. 711-719.
- [4] Tang Xiao-xin, and Chen Guo-hua, "Construction of Enterprise Information Portal in Cigarette Enterprise", International Conference of Management Science and Engineering, 2007, pp. 293-296.
- [5] Stefan Tai, Thomas A. Mikalsen, and Isabelle Rouvellou, "Using Message-oriented Middleware for Reliable Web Services Messaging", Volume 3095 of Lecture Notes in Computer Science, pp 89-104, 2004.
- [6] Birrel A D, Nelson B J, "Implementation Remote Procedure Calls", ACM Transaction on Computer Systems, 2(1), pp 39-59, 1984.
- [7] Han Weihong, Huang zizhong, Wang Zhiying, "Current status and development of object and transaction management technology", Shanghai micro computer, vol 29, pp 23, July 1999.
- [8] Pengwei Zhang, Jingxia Chen, "Research and application of asynchronous message transmission on JMS", Computer Engineering and Technology (ICCET), vol 6, pp 355-358, April 2010.
- [9] A. Y. Halevy, "Answering queries using views: A survey", Very Large Database J., 10(4), pp 270-294, 2001.

- [10] R. Hull, "Managing semantic heterogeneity in databases: A theoretical perspective", In Proc. of the 16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'97), 1997.
- [11] J. D. Ullman, "Information integration using logical views", In Proc. of the 6th Int. Conf. on Database Theory (ICDT'97), volume 1186 of Lecture Notes in Computer Science, pp 19~40. Springer, 1997.
- [12] Frédéric Pourraz, and Hervé Verjus, "Managing Service-Based EAI Architectures Evolution Using a Formal Architecture-Centric Approach", Enterprise Information Systems, 8th International Conference, ICEIS 2006, pp 269~280, May, 2006.
- [13] Mark Mosley, et al., "The DAMA Guide to The Data Management Body of Knowledge (DAMA-DMBOK Guide)", pp 110, 2009.
- [14] Tapati Bandopadhyay, Pradeep Kumar, and Anil K Saini, "A Data-Driven View of the Evolving IT Infrastructure Technologies and Options", Technology Systems and Management, First International Conference, ICTSM 2011, pp. 297–304, February, 2011.
- [15] Javier Espadas, David Romero, David Concha, and Arturo Molina, "Using the Zachman Framework to Achieve Enterprise Integration Based-on Business Process Driven Modelling", OTM Workshops, Lecture Notes in Computer Science, Volume 5333/2008, pp. 283–293, November, 2008.
- [16] John Schmidt, David Lyle, "Integration competency center: An implementation methodology", pp 11, 2005.